Master in Innovation and Research in Informatics

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC)

# Development of Crowd Simulation Models using Unity for Immerssive VR Applications

Master's Thesis

*Author:*
Alicia Nicás Miquel

*Director:*
Julien Pettré

*Co-Director:*
Nuria Pelechano

January 21, 2019

**Abstract**

This project aims to research human behavior in crowds. We use virtual reality in our research as a tool that allows us to perform crowd experiments with great control over the experiment conditions.

We develop a virtual reality application that simulates a lecture-type talk in a crowded room and an emergency evacuation taking place in that environment. We perform an experiment where we immerse five participants in our application. We analyze their behavior when they are immersed in both the emergency evacuation and a normal exiting of the room after the talk.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

I carried out this project at INRIA in Rennes.

The French National Institute for Research in Computer Science and Automation, INRIA is a public research institute for informatics.

The Rainbow team is a joint Inria/IRISA Project-Team in partnership with CNRS, Université de Rennes 1, and Insa of Rennes. Its research is in robotics, targeting applications such as medical robotics, assistive mobility devices, and coordination of multiple robots for spatial navigation tasks.

Julien Pettre is a researcher with the Rainbow team working on crowd simulation and the study of crowd behaviors.

Simulated crowds are required in a wide variety of application areas, from videogames and movies to architectural simulations.

Achieving realism in crowd simulations requires in-depth knowledge about the behavior of humans in crowds. The development of crowd simulation algorithms at the local interaction level requires extending knowledge on individual behaviors in crowds and their repercussion at the collective scales.

Knowledge about the behavior of crowds may be acquired by analyzing observations of real-world crowds or by doing controlled experiments. Hovever, doing crowd experiments with real crowds at even a modest level of complexity is hard to do with real-life crowds, due to problems with standardizing conditions and ethics problems with putting people in certain situations.

Virtual reality allows us to set real humans in interaction with their digital counterparts. Doing so, we can study more detailed situations of local interactions, perfectly control our experimental situations, and more easily acquire experimental data.

This project is part of a longer project aiming to study the behavior of people in a crowd during stressful situations.

In the frame of this masters' thesis, we develop a virtual reality application that simulates a crowded environment and a stressful event taking place in that environment.

We run a small pilot experiment where we immerse five participants in our application. We analyze the data from our experiment in order to validate the usefulness of our application in the larger project.

# Chapter 2

# Related work

This project is primarily concerned with the study of human behavior in crowds. Achieving realism in crowd simulation requires studying crowds in order to understand of how people interact with each other, and how these interactions affect the crowd as a whole.

In Section 2.1, we present the state of the art of the research into the behavior of humans in crowds. We discuss the techniques and methods used to gather insight into this problem, and the difficulties associated with research into any kind of complex question about crowds.

One possible workaround for the difficulty of studying human crowds is using virtual reality. In Section 2.2 we define the concept of virtual reality and present the research into human behavior and specifically crowd behavior that has been done using VR.

In this project, we specifically focus on studying the behavior of the people in a crowd during the course of the evacuation of a building due to some dangerous event such as a fire. In Section 2.3, we present some of the existing research into the evacuation behaviors of crowds.

Finally, for us to do crowd research in VR, we need simulated crowds. In Section 2.4, we present the state of the art in crowd simulation and cpecifically RVO, the algorithm we use in our simulated crowd.

## 2.1 Human Behavior in Crowds

The study of the behavior of crowds is an area with a wide range of applications.

Events that cause the congregation of large crowds, such as sports matches, large concerts, or public demonstrations, have a need for crowd management strategies. These are necessary to prevent disasters and ensure public safety.

The design of public spaces also requires knowledge about the behavior of crowds [66].

Knowledge about the behavior of crowds is also necessary for the creation of realistic crowd simulations, and in turn, simulated crowds are used in crowd management or public space design applications.

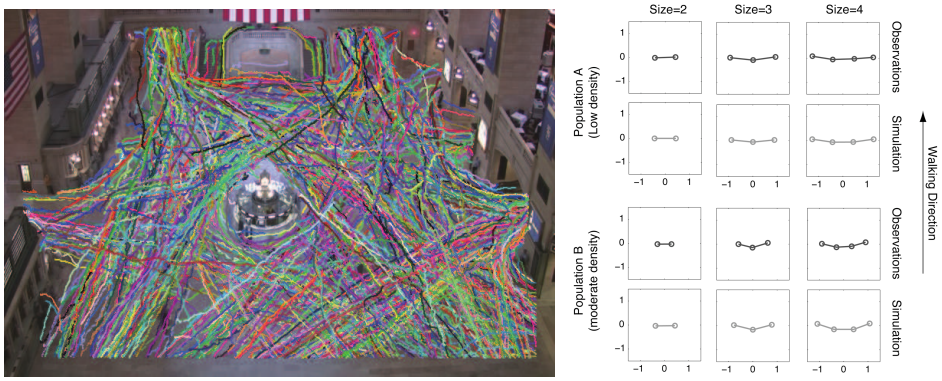
We need to observe human behaviors in crowds to gain knowledge about them. In this section, we present previous work done on observation of crowd behaviors. In Section 2.1.1, we present work done on real world observations, where crowds are observed in their natural state without any intervention from the observer. In Section 2.1.2, we present several controlled experiments done to complete and refine the real world observations.

### 2.1.1 Real World Observations

Real world observation studies observe the behavior of crowds where they naturally happen. The data collected from such studies are free from experimental bias, and these observations can capture a wide range of behavior [13].

We can use these kind of studies to learn about the global behavior of the crowd. There have been many studies that try to analyze global patterns that emerge from the interactions of a group of people with each other.

Many of these studies use computer vision and learning techniques to develop models of crowd behaviors.




(a) (b)

Figure 2.1: Pattern analysis in observation experiments. (a) Trajectories extracted from the observation of a train station [68]. (b) Average walking pattern for groups of typical sizes [45].

Zhou *et al.* [68] learn a model of crowd behavior from the analysis of a

corpus of video from a train station, and complements this learning with the social forces crowd simulation model (see Section 2.4) to improve its robustness. The trajectories they extract can be seen in Figure 2.1a.

Moussaïd *et al.* [45] find that, in contrast with the focus on modeling individuals that is common in crowd simulation methods, about 70% of the people in their observations walk in a group. This and previous studies with similar findings [7] indicate that the study of the dynamics of groups walking together is an important component of learning about crowd behaviors.

Moussaïd *et al.* further analyze the patterns of groups walking together, and find that groups usually walk in an abreast formation, transitioning to a V-shape when the space is tight (see Figure 2.1b).

Peters and Ennis [50] also analyze the adaptation of groups to the environment an further propose a methodology for using a corpus of real-world observations to construct visually plausible crowds.

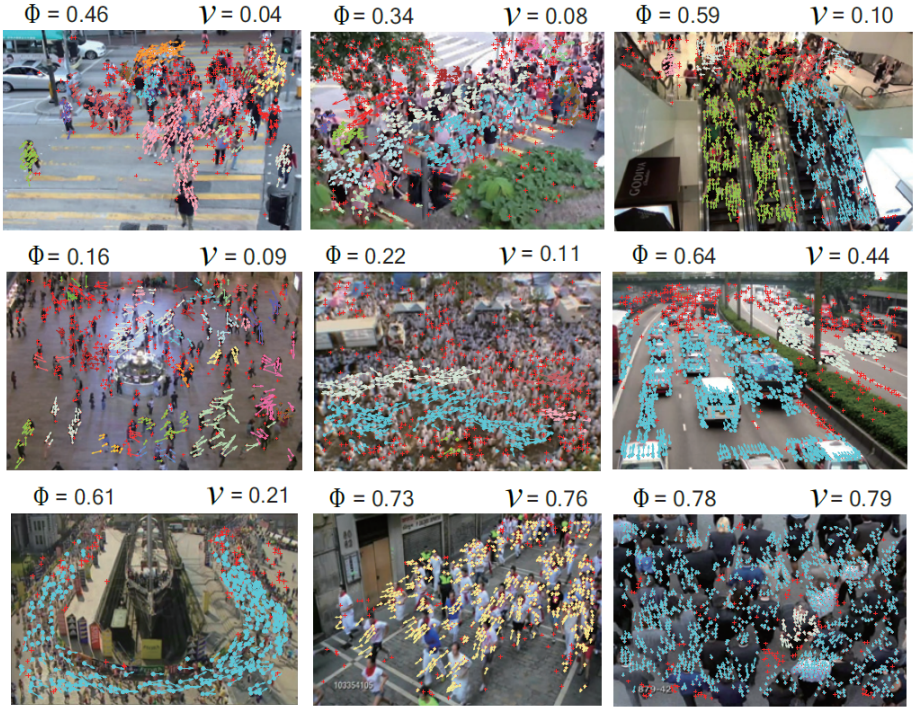



Figure 2.2: Collective motions detected in crowd videos in [67]

Zhou *et al.* [67] propose a measure of the collectiveness of a crowd, i.e., whether the crowd can be considered as a big entity or a collection of individuals. They use this measure in the detection of collective motion in videos (see Figure

2.2. This is important as measurable properties of crowds are needed to perform objective comparison of crowd simulation methods.

Another use of real-world observations is to develop statistical descriptors of crowds. Corbetta *et al.* [16] develop statistical descriptions of traffic in a busy corridor in the form of fundamental diagrams describing the relationship between the density of the crowd and its average speed.

Costa [17] develops statistical descriptions of the distances between members of a group walking together, and additionally observes patterns of group formations, noting for example that large groups often fragment into smaller units of 1, 2, or 3 people.

A third use for real-world observations is to extract trajectories of real people that can then be used for comparison with crowd simulation methods or be reused in simulations [6].

One drawback of real-world observation as a source of data is the lack of control over the experimental conditions. There are many influencing factors and unknown variables that may affect the data in real-world crowds, and it's impossible to control for them all or even know what they are [13].

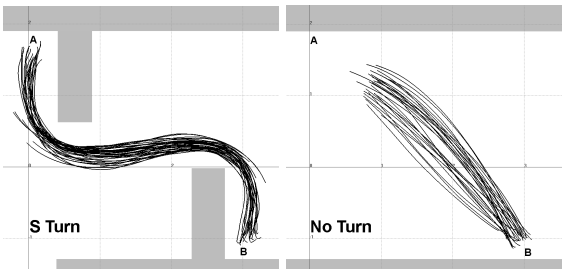
A possible way to overcome these limitations is to run controlled experiments.

### 2.1.2 Controlled Experiments

Controlled experiments can be used to study specific situations in controlled conditions. In contrast with the real world observations of Section 2.1.1, in a controlled experiment the researchers have control over the initial conditions. They also have more knowledge about the precise position and goal of the participants in the experiment, and can know some relevant facts about them, such as their personal characteristics (age, cultural background, etc.). A controlled experiment constrains the observed situation to that specified in the experiment, and may cause the behavior of the participants to be artificial and different from their natural behavior. On the other hand, it allows for the isolation of the specific variable or behavior to be studied.

Some experiments study the interaction between a single human and the environment. While these aren't crowd behavior experiments per se, they are relevant because many crowd simulation algorithms (as we'll talk about in Section 2.4) work by independently simulating the behaviors of the individuals of the crowd.

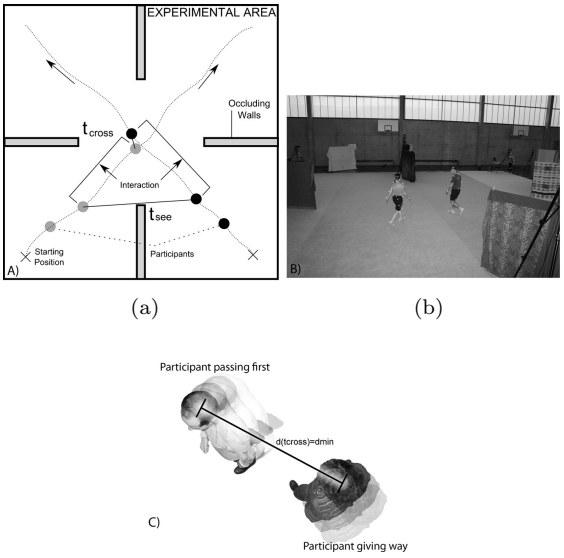
Fajen and Warren [19] develop a model of human trajectories in simple scenes, with or without obstacle avoidance tasks. They find, for example, that when asked to reach a visible goal, their subjects first oriented themselves to-




(a) (b)

Figure 2.3: Observed trajectories from [11] (a) trajectories when an S-turn is forced, showing a constraint on the turning radius. (b) Trajectories with no turn often exhibit curvature

wards the goal and then followed a straight (on average) path to it. Brogan and Johnson [11] study how people trace paths that require different types of turns, and analyze the kinematic and dynamical constraints that influence these paths (see some of these paths in Figure 2.3).




(a) (b)

(c)

Figure 2.4: Collision avoidance experiment from [47] (a) Experimental setup. (b) Picture taken during experiment. (c) *tcross* is the time when the distance between the walkers is minimal.

Other experiments study the interaction between two humans. Pettré *et al.* [51] and Olivier *et al.*[47] study the collision avoidance behavior of two humans that walk in converging trajectories and need to adapt their walking to cross each other while protecting their personal space. Using the experimental setup in Figure 2.4, they find that walkers adjust their trajectories only when there's an actual risk of collision. They also find that collision avoidance is assymetric, with the person who arrives later to the crossing point having to adapt more than the one passing first (see Figure 2.4c). They introduce a measurement, the MPD (Minimum Predicted Distance), of the minimum distance the two walkers will be from each other if they continue walking with their current velocity.

Figure 2.5: Snapshot of the experiment from [37]. Note the formation of lanes in the flow through the bottleneck.

Despite the difficulty of performing crowd experiments in the real world, there are many experiments with more than two participants. Liu *et al.* [37] study the patterns formed by two groups of people moving in opposite directions through a bottleneck of varying sizes (see Figure 2.5 for a snapshot of their experiment). These experiments analyze largely the same things as crowd observation experiments, i.e. they study the formation of patterns in the crowd movements (e.g. stop-and-go waves of movement when walking in line) and statistical values of the crowd (e.g. the flow capacity of bottlenecks of different sizes). These experiments allow more control over influencing factors than observation studies. The increased knowledge about the experimental subjects can also prove useful in the analysis of the data. Chattaraj *et al.* find that the fundamental diagrams of crowds differ between populations coming from different cultures [15].

Doing crowd experiments with a large number of people creates many issues. A big one is the difficulty of managing an experiment with many participants. It's also difficult to limit the effect of non-studied and uncontrolled factors on the experiment [13].

## 2.2 Virtual Reality

Steuer [59] defines virtual reality as "a real or simulated environment in which a perceiver experiences telepresence". The term telepresence, or presence, refers in the virtual reality community to "the subjective experience of being in one place or environment, even when one is physically situated in another" [64][58].

The sense of presence is often evaluated in VR studies using the subjective self-assesment of the user [64][56]. A more objective way of evaluating presence is through measurements of the user's physiological response to events happening in the virtual environment, such as the heart rate response to a stressful situation [41].

The sense of presence achieved by an application has been shown to influence the user's behavior in experiments. Various studies show [57][24] that the sense of presence in the application correlates positively with the subject behaving more like they would in the real world.

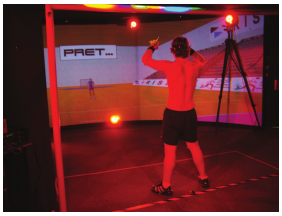


Figure 2.6: Goalkeeper facing a virtual thrower in a handball case study from [9].

Figure 2.7: A burn patient undergoing wound cleaning is immersed in an icy virtual world to distract him from his pain in [30].

The best-known applications of virtual reality are in entertainment, but the properties of presence have enabled VR to be used in the research of human behaviors [9][39][44] (see Figure 2.6 for an example experiment), or as a healthcare

tool [29][36][30] (See Figure 2.7 for an example application).

If we could use virtual reality in our crowd behavior experiments, instead of real-world controlled experiments, it would have some big advantages. First, in a virtual reality experiment, we have full control over the stimuli that we expose the users to. We can make the stimuli interactive and responsive to the user's behavior. We can perform repetitions of the same experiment conditions with a high degree of accuracy. We can also more easily control and isolate the factors that influence our experiment. Finally, virtual reality experiments allow us to study the behavior of the user in the midst of large crowds without the inherent difficulty of designing and coordinating an experiment with a real-world crowd [13].

However, for us to be able to use virtual reality to study human behaviors, we need to know if people behave in virtual environments like they do in reality.

There are in fact several studies that show differences in the subject's perception of their environment in VR and the real world. Banton *et al.* [8] find that users feel that the virtual environment is moving slower than it actually is when they're walking on a treadmill while immersed in the virtual environment. Gérin-Lajoie *et al.* [25] find that the size of the personal space bubble of a person is slightly increased in the virtual environment, while its general shape is preserved. Hollman *et al.* [31] find that walking on a treadmill while immersed in a virtual environment caused instability in the walk of participants. Several studies also find that the perception of distances is compressed in the virtual environment [34][63][53]. Mohler *et al.* [43] however find that this distance underestimation can be reduced after the participant spends 5 minutes walking to targets in a virtual hallway with continuous visual feedback.

On the other hand, there have been studies that show that, while there are quantitative differences in behavior between real-world experiments and their virtual reality counterparts, the qualitative characteristics of the behavior does remain.

[20] compare the obstacle-avoidance trajectories of subjects in the real world and in the virtual environment, and find quantitative differences but qualitatively similar trajectories.

[14] is a study that uses virtual reality for the study of crowd behaviors. They use VR to study how people avoid a large incoming group of people (do they snake through it, or do they go around it) depending on the group's size. The trajectories they observe can be seen in Figure 2.8.

Considering the difficulty and limitations of real-world experimentation and the limitations of learning from uncontrolled observations, we consider that crowd experiments in VR are a useful alternative that can be used to study complex crowd behaviors. The perceptual differences that users experience in

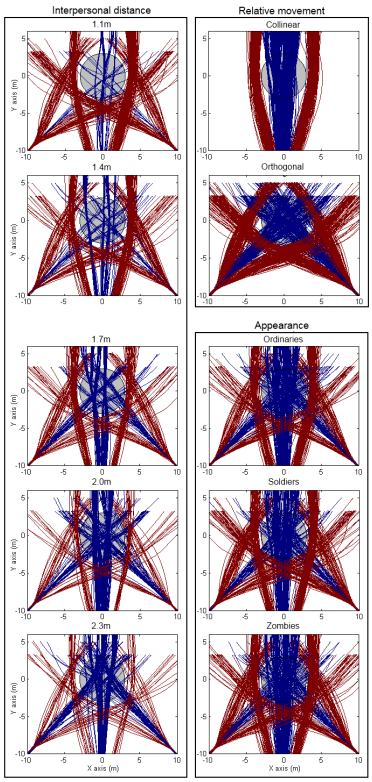



Figure 2.8: Trajectories followed by participants in the virtual environment in [14] wrt the different variables studied (the interpersonal distances of the group, the direction of the movement of the group, and their appearance)

VR experiments should however be carefully accounted for. The need to ensure that presence is achieved in the application if we want the user's behavior to be similar to their real-world behavior should also be considered.

## 2.3 Evacuation Behaviors

We use the word 'stress' to refer to a state of physical of psychical tension that result from factors —stressors— that tend to alter an existent equilibrium [42].

Stress is a wide concept, encompassing acute and chronic stress. There doesn't exist a full consensus on its definition [33]. Some definitions of stress are:

"A particular relationship between the person and the environment that is appraised by the person as taxing or exceeding his or her resources and endan-

gering his or her well-being." [22]

"A substantial imbalance between demand (physical or psychological) and response capability, under conditions where failure to meet the demand has important consequences." [40]

In our project, we consider only acute stress due to an active emergency situation, such as a fire or an earthquake.

These kinds of catastrophes necessitating an emergency evacuation of large numbers of people happen regularly around the world. In studying these evacuations, we regularly find inappropriate behaviors on the part of the evacuees. People may take a while to start evacuating, they may not use the closest exits, or they may push on the people around them, which is counterproductive, as it reduces the throughput of exits, and may make people fall and be trampled by the crowd [12].

It's important to study, and be able to accurately reproduce, the behaviors of people facing an evacuation. Knowing more about these behaviors can help us design buildings that are better prepared to meet the building occupants' needs in an emergency evacuation [35].

During a fire, someone's psychological stress levels may rise because their capacity for processing information is exceeded [52], or they are confronted with an unfamiliar situation [35]. Too much psychic stress can impair cognitive processes and how an individual responds to a given situation [52].

On the other hand, the commonly held idea that there's panic —defined as irrational behavior that may be counter-productive, like pushing people to get through exits— has been found to not hold up. Multiple studies have found no proof of the presence of panic in major disasters [32][26].

In fact, [32] finds that what outside observers perceive from the outside as irrational panic behaviors are described by the person involved in terms of rational decisions, and, additionally, that altruistic behavior is the norm.

## 2.4 Crowd Simulation

Simulated crowds are required by a wide range of applications. See Figure 2.9 for an example application in TV filming. These applications don't all have the same requirements for the properties of the simulations. Videogames and movies require their crowd simulations to be believable while also allowing high control over them. Videogames and interactive experiences additionally require real-time performance and interactable agents. In applications related to event security or urban planning, for example, the main concern is the statistical accuracy of the simulation and its ability to predict the outcomes of the simulated

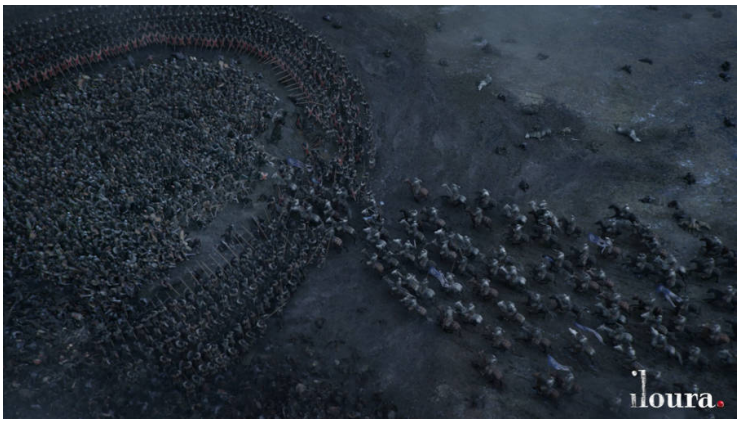



Figure 2.9: A still from Game of Thrones using agents simulated with the Massive crowd simulation software.

situation [65].

The range of models and algorithms of crowd simulation is, accordingly, wide and varied. To a first approximation, they can be divided into two main approaches: microscopic and macroscopic algorithms [13].

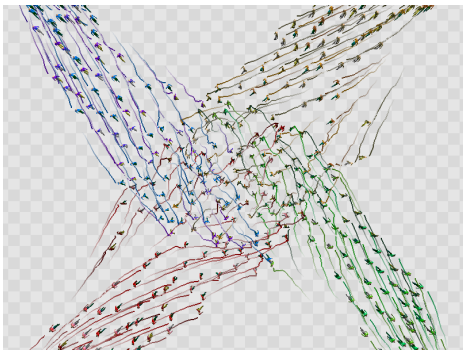
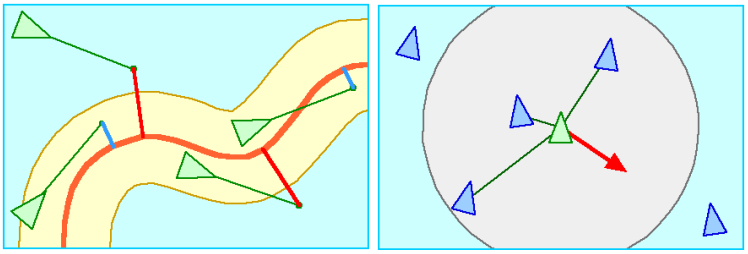


Figure 2.10: Emergent vortex as four groups simulated by [60] cross paths.

Macroscopic algorithms model the behavior of the crowd as a whole, with models being derived from fluid dynamics [55][60] (see Figure 2.10 for some of their results). This kind of algorithms is useful for the simulation of very dense crowds with a small computational cost. However, some issues arise because these algorithms model the agents as particles carried by the flow. The agents can have erratic trajectories and the individual agents can't have the kind of complex interactions that can be modeled with microscopic approaches. This type of algorithm therefore breaks down in the simulation of less dense crowds.

Microscopic algorithms consider each individual member of the crowd independently. They model the behaviors of the individuals, and model a crowd by simulating multiple independent individuals.

Microscopic algorithms focus on modeling local interactions. They model the interactions of the agent with the elements of its environment —mainly obstacles and other agents. These interactions can be of many types. The most common and therefore most studied interaction is collision avoidance.



(a) Path-following rule (b) Separation rule

Figure 2.11: The path-following and separation rules described in [54]

One of the first crowd simulation models, that due to Reynolds [54], models interactions as rules, such as those shown in Figure 2.11. It works by giving each agent a steering vector that determines its direction and speed of movement. This steering vector is modulated by the application of "steering behaviors" that result from the evaluation of the rules of the model. For example, if the agent is too close to a nearby agent, a steering force will be added to its steering vector that steers it in the direction away from the nearby agent.

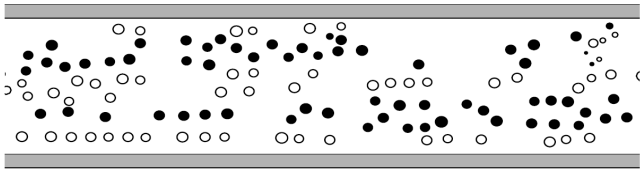


Figure 2.12: The formation of lanes in a crowded hallway modelled by [27].

The social forces model [27] models individuals as particles subjected to various forces. The motion of the agents is then defined by a sum of forces, where each of the forces corresponds to an interaction. The model can reproduce emergent patterns such as the lane formation shown in Figure 2.12.

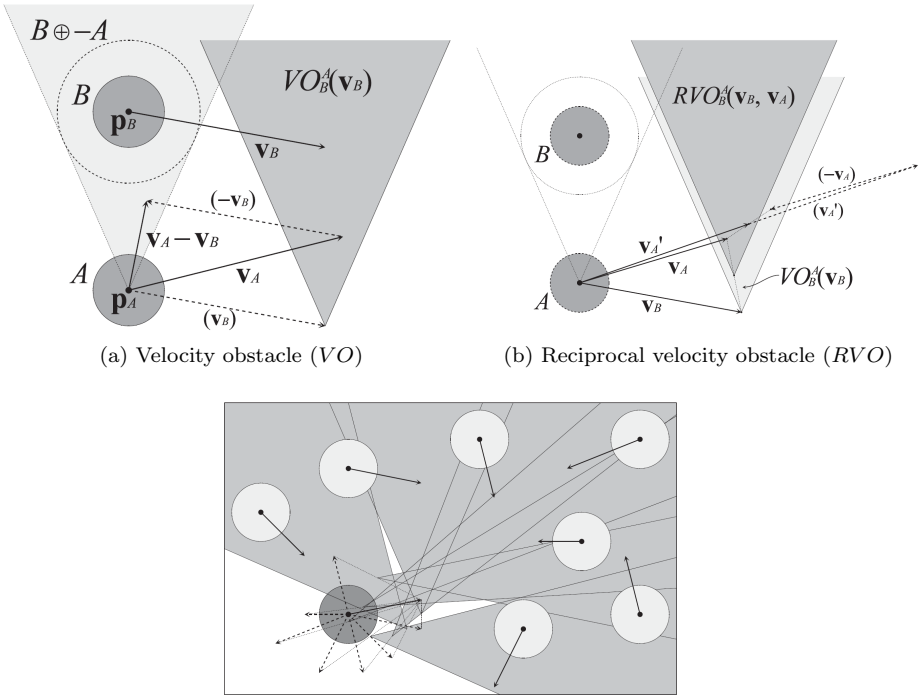
These are position-based algorithms, where the next position is computed by adding forces to a steering vector. They are mainly reactive, they only

modify the behavior of the agents when there's a risk of collision. Another set of microscopic algorithms try to predict future risk of collision and base the behavior of the agents on it.

Velocity-based algorithms, such as RVO [61] compute the agent's next velocity by adding geometric constraints to the space of possible velocities. They then choose the best velocity, according to some metric, among the remaining ones.

In Section 2.4.1, we describe RVO, the velocity-based microscopic algorithm that we use in our crowd simulation.

### 2.4.1 RVO




(a) Velocity obstacle ($VO$)

(b) Reciprocal velocity obstacle ($RVO$)

(c) Union of all the reciprocal velocity obstacles the agent will avoid

Figure 2.13: Reciprocal velocity obstacles as computed in [61]

In our crowd simulation, we use RVO to model the collision avoidance behavior of the agents in the crowd.

Reciprocal Velocity Obstacles or RVO [61] is a microscopic, velocity-based, collision avoidance algorithm. For every simulation step, RVO computes the ve-

locity of an agent in the next simulation step by applying geometric constraints to the space of possible 2D velocities. The set of discarded velocities is known as the agent's velocity obstacle. Once the constraints are applied, it chooses the new velocity as the remaining velocity closest to the agent's preferred velocity. If no permissible velocities remain, a velocity inside the velocity obstacle that has a high time to collision may be chosen. RVO considers the velocities of surrounding agents when computing the velocity constraints. This allows the algorithm to act on the future risk of collision and better handle collision avoidance with agents or obstacles moving at high speed.

RVO is based on an earlier algorithm, called Velocity Obstacles [21]. As an example, let's consider the computation by the VO algorithm of the velocity obstacle of an agent $A$ induced by the presence of an agent $B$. Figure 2.13 shows a graphical representation of this computation. VO computes the velocity obstacle $VO_A^B(v_b)$ as the set of velocities $v_a$ that will lead to a collision with the agent $B$ moving at a velocity $v_b$ at some point in time. The velocity obstacle is computed in the VO algorithm as the set of velocities $v_a$ such that the relative velocity between the two agents $v_a - v_b$ intersects the Minkowski sum of $B$ and $-A$.

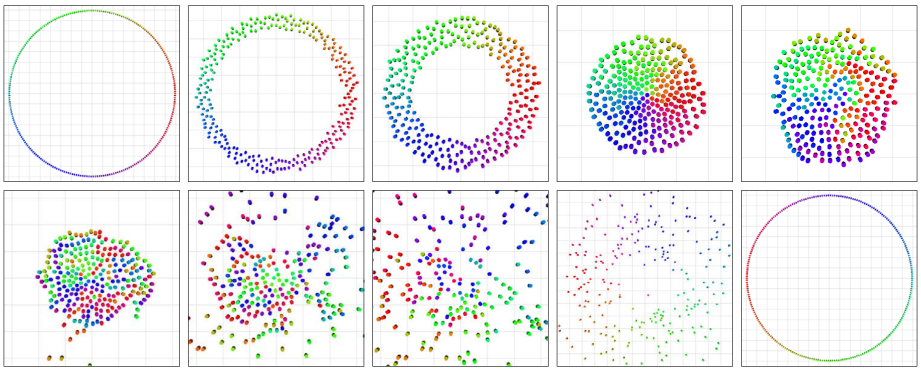


Figure 2.14: RVO [61] simulation where 250 agents move to the diametrically opposite positions in a circle.

RVO is an extension of VO that is "reciprocal" because each agent assumes that other agents will try to avoid it as much at it will try to avoid them, and therefore does only half of the necessary collision avoidance. This removes the oscillating velocity artifacts caused by the VO algorithm.

Figure 2.14 shows some results obtained with RVO.

There are some situations where RVO leads to artifacts. In particular, in dense situations, several agents may get stuck in a symmetrical pattern, where none of them is able to get the right of way. [BRaVO: Biased Reciprocal Velocity

Obstacles Break Symmetry in Dense Robot Populations]

# Chapter 3

# Overview

This project is part of a longer project. The ultimate goal of the larger project is to study the behavior of people in a crowd during stressful situations. For the reasons explained in Section 2.2, we consider that virtual reality is a useful tool to reach that goal.

In the frame of this masters' thesis, we do two main tasks. First, we develop a virtual reality application. The aim of this task is to create the virtual environment where we will later immerse the participants in our experiments. The application must then simulate a crowded environment and a stressful event taking place in that environment.

We choose to simulate a lecture-type talk (we use a TED Talk as the talk audio) in a small venue with a crowd of 100 people attending. We script and develop two possible endings to the talk. In one of them —our baseline scenario— the talk ends normally and the crowd leaves the room calmly. In the second scenario —our emergency scenario— the talk is interrupted by an explosion and the subsequent emergency evacuation of the venue.

In Chapter 4 we describe the development process of the application. In Section 4.1 we describe the pre-existing tools we built our application on top of. Section 4.2 describes some technical aspects of the development of our application qua game-like application. Section 4.3 describes some challenges we encountered regarding crowd simulation, and the solutions we arrived at.Finally, Section 4.4 talks about the structure of the code developed for the application.

In the second part of the project, we do some preliminary validation of the application we built. Our goal with this validation is to figure out if it's suited to our research goals. Since we want to study stressful situations, we want to know if the application (specifically, the emergency scenario) is able to induce stress on the user. Because we want to study behavior, we want to check whether our

application and the data we collect can be used to measure a change in behavior between our two scenarios.

Chapter 5 describes our pilot experiment in detail.

Chapter 6 describes the analysis we do on the data from our experiment. Section 6.1 looks into the stress question using physiological data to detect stress in the participants. Section 2.1 looks into the behavior of the participants, examining the spatial data of the positions and velocities of the participants, as well as their gaze directions.

Finally, Chapter 7 describes our conclusions and the future work that needs to be done as the continuation of this project.

# Chapter 4

# Development

In this chapter, we detail the work that went into the development of our VR application.

## 4.1 The Tools

Our application was built on top of several tools (software, hardware, game assets) that made development considerably faster and easier. We describe them here.

### 4.1.1 The Game Engine

Unity is a widely-used cross-platform game engine that has a wide range of features for graphics and game development and good VR support. Here, we offer a short overview of some of the features we used in the development of our application.

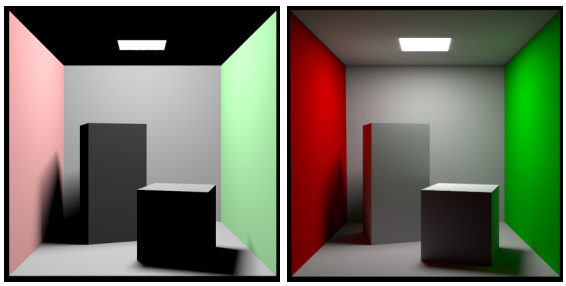
**Lighting and rendering.** When a source of light illuminates a scene, that light is bounced off of the surfaces in the scene onto other surfaces. Light that comes from bouncing on a surface instead of directly from the light source is called indirect light. A lighting system that models the effects of indirect light on the scene is called a global illumination (GI) system [2]. See Figure 4.1 for a comparison of global illumination and only direct illumination.

Traditionally, graphics applications have been constrained by computational resources to only computing direct illumination. Still now, lights that must be computed in realtime are mostly constrained to direct lighting.

Unity provides three main lighting techniques: realtime direct lighting, baked GI lighting,and precomputed realtime GI.

Realtime lighting has the advantage of flexibility. Because they aren't precomputed, realtime lights can move around the scene or change their color and intensity during gameplay [4]. However, the computation of global illumination is too expensive to be done in realtime, so realtime lights only offer direct illumination.

Light sources that are baked have their effects on the scene pre-computed and stored before runtime. Unity provides a system of baked global illumination that can compute indirect lighting, but, because it is precomputed, the properties of lights that use this system can't be modified during runtime.



(a) (b)

Figure 4.1: (a) Render with direct lighting only. (b) Render with global illumination. Indirect lighting computation is needed for effects such as color bleeding.

The third system, precomputed realtime GI, offers indirect lighting and therefore higher realism than direct lighting while allowing light properties to change during runtime. It works by precomputing the surface-to-surface bouncing of light in the scene. The last segment of light bouncing from the light to the first surface it hits is then computed in real time, and the effect is then propagated through the precomputed bounces. One disadvantage of this system is that computation is spread out over a few frames because it's expensive. This means that changes in light properties take a few frames to propagate through the scene. The expense of the computation can also be a problem in computers with low resources [4].

We use a mix of baked and precomputed realtime GI, for reasons explained in 4.2.2

**Pathfinding** Unity has support for some of the elements needed to build a crowd simulation. It includes support for pathfinding (the problem of finding a path in a scene between two points) in the form of a navigation mesh implementation.

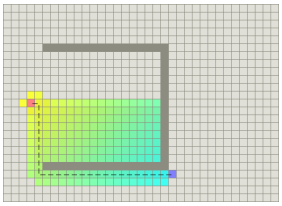


Figure 4.2: A* pathfinding on a regular grid. Colored nodes are the ones that have been explored

A navigation mesh works by dividing the walkable area of a scene into a set of convex polygons connected to each other in a graph. Navigation between two nodes in the polygons can be done with graph search algorithms, such as A* (see Figure 4.2). Navigation inside a polygon can be done in a straight line, since the polygon is simple and convex [62].

Unity implements RVO [3] as an obstacle avoidance algorithm for agents travelling the navigation mesh. We use our own implementation of RVO instead, which allows us to tweak it to suit our needs better.

**Mecanim** Mecanim is Unity's humanoid animation system. Features of it are easy retargeting of animations, a state machine system for specifying behaviors, and blending between animations by linear interpolation, in time and concurrently.

Animation retargeting refers to the ability to apply animations from one character to another. Retargeting can be done between humanoid characters with relative ease because all have the same general structure of limbs and joints. Figure 4.3 shows the bone structure that a model must have to be compatible with mecanim.

### 4.1.2 The Experimental Platform

**CrowdMP** CrowdMP is a platform for developing crowd simulation experiments with Unity. It's a team project, started 5 years ago, which has had several contributors over the years and is currently being maintained mainly by Julien Bruneau.

CrowdMP provides some features that are useful when running experiments, like user controls, data recording, support for the management of trials and for

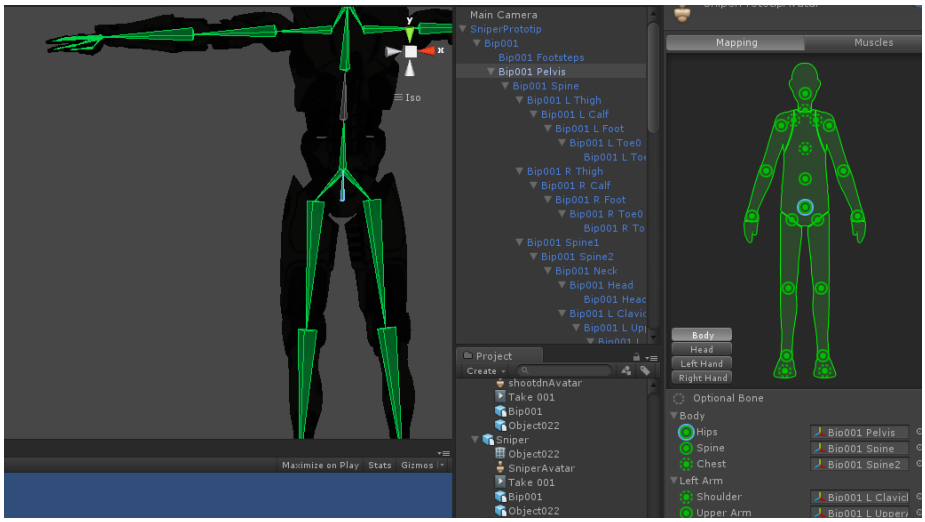



Figure 4.3: The bone structure of a humanoid in mecanim.

the display of instructions for the participants, etc. It also offers features specific to crowd simulation. It comes with a variety of character models and animations, and the ability to configure the behaviors of the members of the crowd (control laws, simulation algorithms, etc.) to suit the experiment. It provides add-ons that support integration with several virtual reality platforms, including the FOVE head-mounted display. Additionally, it allows to easily extend it with new features or plug-ins.

**FOVE** FOVE is a virtual reality head-mounted display that provides eye-tracking capabilities. A head-mounted display (HMD) is a VR platform that allows for a high level of immersion. HMDs cause simulator sickness in users more often than other immersive VR systems, which can negatively affect the sense of presence in those affected [64].

FOVE is an HMD with a 100-degree field of view (comparable to other widely used HMDs, like HTC vive and Oculus Rift, and a high screen resolution of 2560 x 1440 pixels [23].

Part of the goal of the project is to study the gaze behavior of people. We use the FOVE headset eye tracking to collect the necessary gaze data.

**RVO** CrowdMP allows the use of RVO2 as a simulation algorithm for the crowd. RVO2 is an extension of RVO that replaces the RVO framework with the optimal reciprocal collision avoidance (ORCA) algorithm. This change is meant to increase the computation speed, the smoothness of the motion of

agents, and to guarantee collision avoidance between agents [1]. CrowdMP provides an interface to the C# implementation provided by the authors, which is available at `https://github.com/snape/RVO2-CS/`

**Rocketbox** The agent models used in the experiment are from Rocketbox, which has a large set of highly realistic models of people.

**Mixamo Animations** Some animations were taken from Mixamo to complement the animations already in CrowdMP. Mixamo has a large library of animations aimed at game development. The animations are compatible with Unity's Mecanim system and can be easily integrated with the existing animation in CrowdMP.

## 4.2 Technical Aspects

### 4.2.1 Sound

Realistic sound is an important component of achieving a sense of presence in a virtual reality application [28].

3D audio is audio that is manipulated such that it appears to come from a specific position with respect to the listener's ears. Unity allows the developer to create a 3D sound source by simply positioning it in the scene.

Additionally, Unity offers a real-time mixer, that allows to test and modify the sound level of the different sound sources at runtime, and a library of sound effects.

The sound for the performance is taken from a TED Talk. The rest of the sounds are taken from `freesound.org` and the BBC library of sound effects and field recordings (`bbcsfx.acropolis.org.uk`).

We used 3D sound for the on-stage performance. The rest of the sounds — the noise of the crowd, the explosion, the emergency sirens and the evacuation notice— are regular 2D sounds, since they come from all around the listener.

We added reverberation sound effects to the noise of the crowd (quiet talking during the performance, sounds of the crowd leaving the performance), and a low-pass filter to the sound of the ambulance sirens and screams so that they appear to come from outside the room.

### 4.2.2 Lighting

Rendering lights is a computationally expensive part of our application. The nature of our application requires rendering a lot of fairly complex models in

most frames.

Some requirements due to the nature of our scene create constraints on the lighting that preclude using less expensive solutions. We need the lights to change when the performance ends and when the explosion happens. This makes baked GI not a possibility for the main lights, which means that we need to use realtime lights for the main lights in our scene.

We use Unity's precomputed realtime GI to achieve greater realism than regular direct lighting. Dynamic objects (i.e. the crowd) are lit using light probes. Light probes are sample points situated in the scene that store the light hitting that point from all directions.

The lights that light the rooms outside the main room are baked lights. Mixing precomputed realtime GI and baked lights adds some overhead, as does adding more realtime lights to a scene. We don't see a performance difference between the two approaches for the outside lights.

Another issue has to do with per-pixel vs. per-vertex lights. Per-pixel lights compute lighting at each lit pixel, while per-vertex lights do it only at each vertex. Per-pixel lights cause a big performance overhead and impede certain optimizations. In our scene, the lights on the stage should be rendered per-pixel because they are simulating spotlights on a performer.

To reduce the number of lights on the stage to two, while maintaining the illusion of a multi-light stage lighting rig, we use a single spotlight with a cookie (a texture added to a light to change its shape) to fake a multi-spotlight wash. However, two per-pixel lights are still somewhat problematic, especially because they illuminate the crowd too. We don't try to optimize this further, because the computer we use to run our experiment is powerful enough.

The use of shadows would be a big improvement on the realism of the scene, but they imply a big performance overhead.

The appearance of the emergency exit signs glowing in the dark room is simulated with the bloom post-processing effect.

## 4.3 Crowd Simulation

In this section, we talk about some aspects of the simulated crowd in our application.

Our simulated crowd needs to look like a crowd at a performance. During the performance, they should stay in place and look at the stage. When the performance ends, they should applaud, and then leave in an ordered way that looks as realistic as possible. In our emergency scenario, when the explosion happens, the crowd should look startled and scared, and then leave in a hurry.

Our crowd simulation is a microscopic crowd simulation, i.e. each agent is simulated independently, and a crowd is made up of a group of independent agents.

The locomotion of each agent of our crowd is made up of two parts: a control law, which handles pathfinding, and a simulation, which handles local movement (in this case, the only local movement behavior we handle is collision avoidance). Each agent additionally has animations. The animations are controlled using Unity's system of humanoid animation.

We use RVO as a local movement algorithm. An overview of RVO can be found in Section 2.4.1. During development, we encountered some situations in which the way that RVO does collision avoidance led to unrealistic behaviors. These issues and the solutions we found are described in Section 4.3.1.

Section 4.3.2 describes the way we handle pathfinding in our simulation as well as some issues we had with the exit doors as crowd flow bottlenecks and how we solved them.

Section 4.3.5 describes how the animation of the crowd agents adapts to their movement.

### 4.3.1 RVO

In order to use RVO to create a realistic crowd simulation in our application, we needed to address some issues with the algorithm.

Because of the geometric nature of the algorithm, agents sometimes get stuck in a symmetric pattern where all of them yield to the others and nobody moves. In our scene, this happens at doors, where the agents get stuck in an arch pattern (see Figure 4.5a).

When many agents want to go through a door at once, some of them start walking away from the door (see Figure 4.4), since that's the best way to avoid collision. In an evacuation, this looks unrealistic, since what we'd expect is queueing behaviors.

In the next paragraphs, we describe the tweaks we did to RVO2 to prevent these behaviors.

**Forbid walking backwards** by adding an ORCA line that removes velocities with a backward component from the possibilities.

In spite of this fix, the agent may still end up with a small backwards vector because of RVO2 needing to relax some constraints or because of numerical issues. To deal with this, when the simulation asks the agent to move in a direction away from its goal with a very small speed, the agent does move but keeps its previous rotation.

Figure 4.4: Agents walking away from their goal.

**Add a random tie-breaking component.** Each agent is given a random value that represents how willing they are to squeeze through in a tie situation.

(a) (b)

Figure 4.5: (a) An arch pattern at a doorway. (b) An agent (red) reducing its radius to attempt to squeeze through a crowded area

When an agent computes its best feasible velocity, it uses its radius and its neighbors' radii. Our modification allows the agent to consider its own radius to be smaller in tie conditions. This simulates the behavior that can be observed in a crowded area —such as a crowded subway— where people may try to squeeze through the crowd to be able to move.

To detect a tie condition, the agent checks the time $t_{stuck}$ since it's moved (only if its preferred speed is larger than zero) and reduces the radius it considers itself to have once $t_{stuck}$ reaches a threshold $t_{squeeze}$. $t_{squeeze}$ is an attribute set randomly for each agent and therefore acts as a tie-breaker. Figure 4.5b shows the results.

**Agents get stuck at walls.** When agents simulated with RVO2 encounter an obstacle that runs perpendicular to their preferred velocity, they may be unable to overcome it. This is because the vector closest to their preferred velocity in this case is the null vector. In our scene, agents would get stuck at walls when they tried to exit the room (see Figure 4.6a).


(a) (b)

Figure 4.6: (a) Agents stuck against walls. (b) Invisible walls.

We address this limitation by adding invisible walls to the scene that add a bevel to the inside of the jambs of doorways, as shown in Figure 4.6b. This breaks the perpendicularity of the wall and the agent's preferred velocity and makes it so that the agent can find a valid velocity at which to move on.

Using smarter pathfinding also helped agents not get cornered against walls in the first place.

### 4.3.2 Pathfinding

Our initial pathfinding solution was a basic control law which aimed the agent towards specific waypoints. Setting a waypoint at a doorway and another outside the room gave us reasonable behavior. However, this, as can be expected, led to a few artifacts. We set a tolerance to the waypoints, such that agents could move on to the next waypoint even if they couldn't reach the exact way-

point position. However, when large jams happened at one of the doorway, some agent would exit through an unclogged door, then try to re-enter through its assigned doorway. Aside from unrealistic movements, this caused agents to meet head-on while trying to move in opposite directions, which is a situation that may cause issues with RVO2.

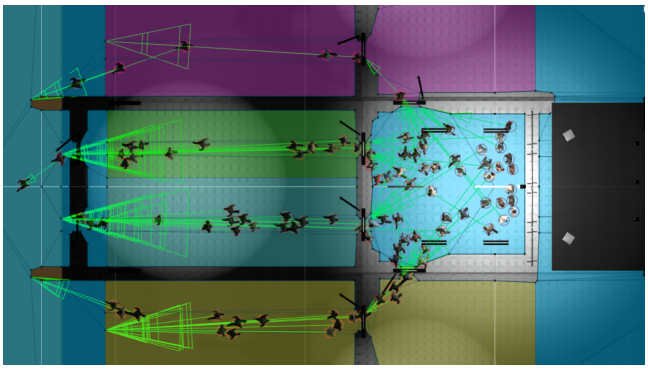


Figure 4.7: Navigation mesh computed for our scene. Different colors represent different areas. The green arrows indicate the agent's current goal.

In order to fix this we switched to Unity's implementation of the navigation mesh algorithm for pathfinding.

Unity's NavMesh navigation system allows you to create characters that can intelligently move around the game world, using navigation meshes that are created automatically from your Scene geometry.

It allows baking a NavMesh, which we use to create the general navigation areas, and it also allows dynamic obstacles, that can move at runtime. We use dynamic obstacles in the emergency evacuation scene, to have casualties lying on the ground, which aren't agents or static obstacles, since they have animations that may cause them to move around a bit.

The NavMesh also offers the possibility to set a per-agent mask to block some walkable areas, which we use to solve the problem of uneven flow in the next section. Figure 4.7 shows our navigation mesh and the distribution of walkable areas we use.

### 4.3.3 Flow Control

Another problem we encountered, is that static ways of assigning an exit door to an agent ended up with large accumulations of people trying to go through a doorway while other doorways were empty. While this is a behavior that is found sometimes in crowds, it led to unrealistic locomotion in the agents waiting

to exit.

We fixed this by setting up a system of subscription to doors. Each agent, when it wants to leave the building, has to find a door with a sufficiently low number of subscribers, and can only exit when it finds one. When the agent has exited the building, it unsubscribes from its door. When a door has a low subscriber count, it sends an event, that all the agents are listening to. The listening agents may then choose to switch their subscription to the low-subscribers door, if its current door has a high amount of subscribers and it isn't too close to leaving already.

To make this work with the NavMesh, we setup different NavMesh areas on each doorway, and block the agent (with the walkable area mask) from walking on areas that contain doorways it isn't subscribed to.

### 4.3.4 Staggered Leaving

The agents in a crowd don't leave all at the same time.

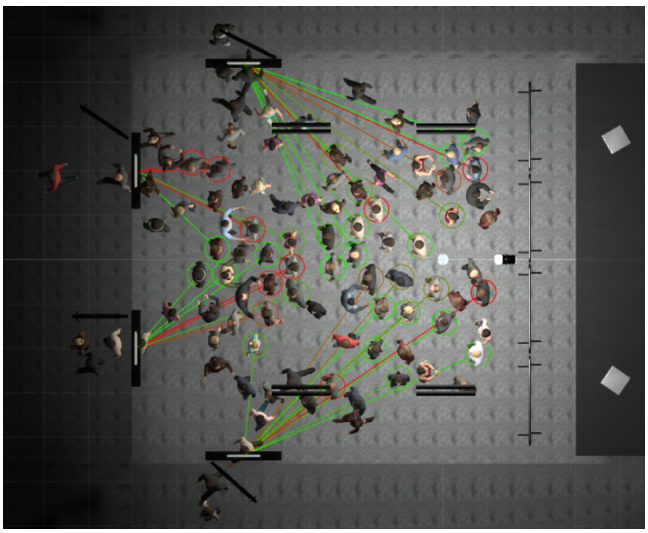


Figure 4.8: Lines showing the ray cast to check whether the agent can start to move. Red means can't move, green means can move.

They cast a ray between their position and the door they want to exit through and compute a score based on the number of obstacles that the ray intersects, their distance to the agent, and the amount of time that they've been waiting to exit. Given a set of ray intersection distances $\mathbf{r} = \{r_1 ... r_n\}$, and the distance to the doorway $d$ the leaving score at time $t$ since the agent starting trying to leave the room is computed, heuristically, as:

$$leavingScore = 5 - \sum_{r_i \in \mathbf{r}} (\frac{2}{r_i^2}) - min(0.1d, 1) + min(0.2t, 1)$$

Figure 4.8 shows the resulting value for a variety of situations.

### 4.3.5 Animation

Unity offers an animation system called Mecanim. The Mecanim system is a powerful way of setting up animation of characters in games. It offers features like humanoid animation retargetting and the ability to set up the animation of a character with a visual state machine editor that can be extended with C# code to create complex transitions and behaviors. Figure 4.9 shows the state machine for our crowd agents. An issue I encountered often when working with Mecanim, on the other hand, is the lack of documentation of some parts of the Mecanim scripting API.

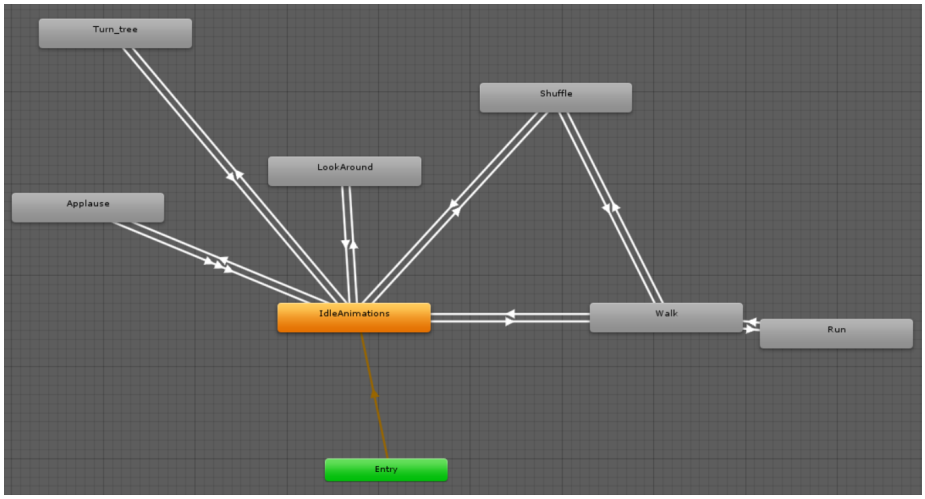



Figure 4.9: The animation state graph of the agents.

**Walking and Running** Unity offers the possibility of blending several animation clips into one animation via a feature called blend trees. Blend trees are a good way to set up walking and running animations that change according to the angular velocity at which the agent is moving.

We have animations of people walking forward and walking in wide and narrow circles, right and left. Using a blend tree allows us to create a continuum of animations between our existing animations by linearly blending them and

using some parameter set programatically to set the blending weight. The blending weight in our walking and running blend trees is set as the angular velocity of the agents.

The agents choose to use walking or running animations depending on their speed. This could also be set as a blend tree parameter, creating a 2D blend tree with axes speed and angular velocity, but the results weren't visually convincing.

**Shuffling** When their speed is low enough, normal walking animation slowed down starts looking unrealistic. Instead, they use an animation where they do one step, then stand for a bit, then do another step.

**Turning Around** Sometimes, the agents have to turn around in place. We use animations where the agents turn around in place.

We wanted to keep the animation behavior separate from the locomotion behavior. The turning around animation is triggered when the agent turns around (its forward vector changes direction by more than $45^{o}$ from the previous frame) while its speed is zero. When this happens, the orientation of the root bone of the agent is decoupled from the orientation of the agent object (which is used for locomotion computations) such that the agent appears to continue to look in the same direction. Then, the root bone is gradually rotated until it matches the agent object orientation, while the turn in place animation plays.

## 4.4 Code

Figure 4.10 shows a diagram of the overall code structure of the project.

The code is divided into two main blocks, coordinated by the `IndoorSceneTrial` class, which implements the `TrialManager` class from CrowdMP.

One of the blocks, which we'll call the Crowd block, handles the behavior of the crowd agents. The second block, the Scenario block, handles the "script" of our two scenarios (baseline and emergency).

`MultiControlLawAgent` is the main class in the Crowd block. It behaves as a state machine. `IndoorSceneTrial` communicates the changes in the scenario state to the `MultiControlLawAgent`. The state of `MultiControlLawAgent` mainly follows the scenario state changes with some delays to avoid the unrealistic situation where everyone in the crowd does something at the exact same time.

`MultiControlLawAgent` implements the `Agent` class from CrowdMP. A CrowdMP `Agent` has a `ControlLaw` component (i.e. a pathfinding algorithm) that it queries at each frame to find its next position and rotation. The position and

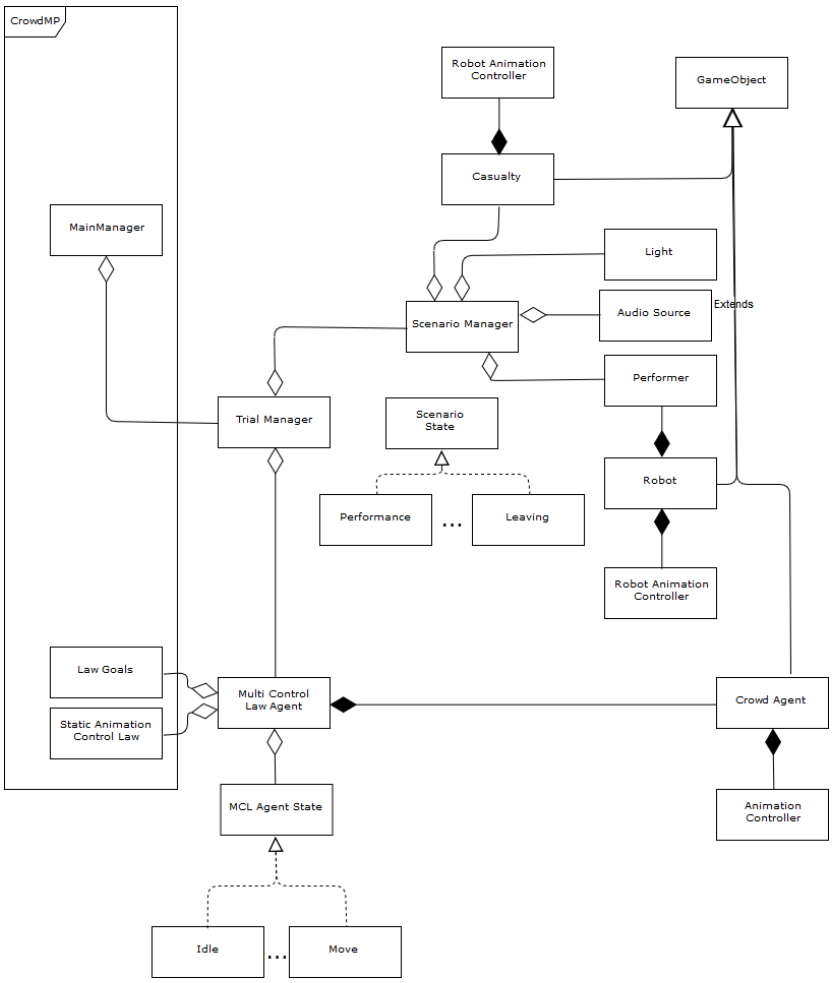



Figure 4.10: The overall structure of the code of our application

rotation given by the Control Law can be overriden by the control simulation (in our case RVO) to handle local movement. `MultiControlLawAgent` extends `Agent` by allowing an agent to have multiple control laws. This is needed so that our agents can start the simulation standing in the same spot to watch the performance, and then start to move towards a goal (the outside).

Each agent object has an `AnimationController` component that manages the animation of the agent independently of its locomotion. At each frame, the `AnimationController` computes the velocity and angular velocity of the agent, plus gets some state information from `MultiControlLawAgent` (e.g. is the agent applauding?) and updates the animation accordingly.

The Scenario part of the code is tasked with coordinating what we could call the "script" of the scene. The `ScenarioManager` class receives from `IndoorSceneTrial` (which in turn is interpreting the instructions from the trial configuration file) the signal to trigger an event (which can be the normal end of the performance or the emergency ending). The `ScenarioManager` class then coordinates the behavior of the elements of the scene not related to the crowd (the sounds, the lighting, the robot performer, and the casualties in the emergency scenario). After the inputs from `TrialManager` have triggered an ending "script", the `ScenarioManager` coordinates the subsequent stages of the script using a timer and constants coded into the class that indicate how long each stage should last. `IndoorSceneTrial` also reads the scenario state from `ScenarioManager` and passes it on to the `MultiControlLawAgent` instances, which use it as part of the input for their own state transitions.

# Chapter 5

# Experiments

With this project, we are doing some preliminary work in preparation for further experiments. The ultimate goal of the larger project is to study the behavior of people in a crowd during stressful situations.

In the context of this project, we run only a small pilot experiment. Our experiment aims to validate the appropriateness of our application to the task.

**Participants** For this pilot experiment, we recruited five participants (ages 25, 27, 28, 31 and 27) three male and two female. The participants didn't have prior knowledge of the goals of the experiment. Some of them had previous experience in VR or videogames, and some didn't.

**Apparatus** For this experiment, we use the FOVE headset, seen in Figure 5.1a as an HMD VR system. This headset allows us to collect data on the gaze direction of the wearer. We use on-ear headphones to enable the user to hear the 3D sound.

The user can navigate through the virtual scene using a joystick (see Figure 5.1b. The longitudinal axis of the joystick controls speed using a function by parts that increases more steeply at high values of the axis position. The lateral axis controls the angular speed linearly. When the joystick is in rest position the speed and the rotation speed are of $0m/s$

The participants were seated on a chair in the experiment room. They were immersed in the scene in a first-person perspective, where they were standing in a room simulating a small concert or event hall and could walk around freely.

We outfitted the users with galvanic skin response and pulse sensors (see Figures 5.1c and 5.1d) to collect physiological data.


(a) FOVE

(b) Joystick

(c) Pulse sensor

(d) GSR sensor

Figure 5.1: Equipment

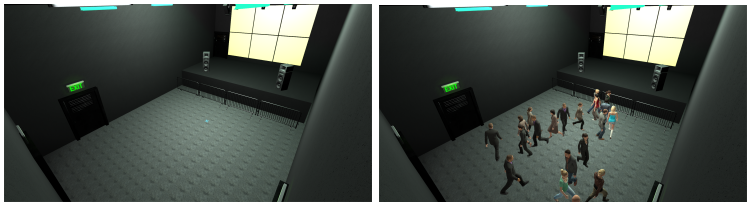
**Task** The task is described to the participants via a slide presentation with slides and text. We ask the participants to listen to a speech in a small crowded concert room, and to exit the room with everyone else when the speech ends. The presentation also explains the usage of the joystick.

For the first part of the task, the participants are asked to perform some preliminary tasks. We include these tasks in order to allow the participants to get used to the virtual environment and the joystick navigation. We didn't record the data from the training.

For the first part of the training, the participants train with the joystick in the concert room emptied of people (see Figure 5.2a. They are asked to walk around the room and out and back in the doors. They are also asked to test the limits of the joystick speed. The second part of the training takes part in the concert room again. This time the room contains 15 virtual agents who walk around following paths between randomly assigned waypoints (see Figure 5.2b).

At the end of the second trial, calibration of the FOVE gaze tracking system is performed.

For the main part of the task, the participant is immersed in the baseline scenario three consecutive times, and in the emergency situation one final time. They are not told about the existence of the emergency situation beforehand.



(a) First training scene (b) Second training scene

Figure 5.2: Experiment training scenes

The user can start each trial themselves by pressing the trigger button of the joystick, and each trial automatically ends when the user exits the room.

After the experiment, the participants are offered gummy candy and nougat.

For each participant and each trial, we record the pulse of the participant, their galvanic skin response data, their gaze direction, and the position and orientation of the participant and all the crowd agents.

**Conditions** The virtual environment consists of a small concert room, with a stage in the front and standing room for the audience.

There are four exits in the room. All of them are marked with a lit emergency exit sign on top. Two of them are in the back wall of the room, and there is another in the back end of each of the side walls. Their distribution can be best seen in Figures 5.5c and 5.3.

The initial position, looks, and behavior of the agents in the crowd are fully defined in the configuration file for each trial. The movement of the agents is controlled by an RVO2 simulator. The parameters of the agent simulations are the following: `neighborDist: 0.8m` (maximum distance at which the neighbors are considered for avoidance), `maxNeighbors: 6` (maximum number of neighbors that can be considered for avoidance at one time. If more than 6 neighbors are within `neighborDist` of the agent, the closest 6 are avoided), `timeHorizon: 2` (maximum time to collision at which an agent is avoided), `timeHorizonObst: 0` (maximum time to collision at which an obstacle is avoided), `radius: 0.28` (the radius of the agent), `maxSpeed: 1.2` (the maximum speed the agent may reach to avoid an obstacle). The maximum speed is raised to 5 at the start of the emergency evacuation. The agents also have a randomly assigned preferred speed.

At the beginning of each trial, there's a crowd of 100 agents in the room. The user starts at a position towards the front of the crowd (See Figure 5.3). There's also a performer on the stage, in the form of a robot (see Figure 5.4).

Figure 5.3: Initial distribution of the crowd. The red circles indicate the radius of the crowd agents. The blue crosshairs indicates the initial position of the agent.

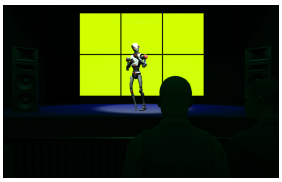


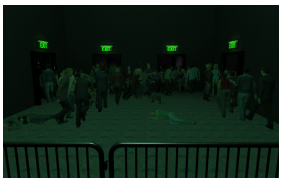
Figure 5.4: The robot giving its speech on the stage

(a) Watching the performance

(b) Applauding crowd

(c) Normal exit

(d) Scared crowd



(e) Emergency evacuation

Figure 5.5: The crowd at various points in the scenarios

Each trial begins with the robot giving a speech. The speech given consists of the last 25 seconds of the TED Talk "Forget multitasking, try monotasking"

by Paolo Cardini. The talk is in English. The experiment participants aren't native English speakers, but most understand English. While the speech is happening, the lighting in the room is dim, while the stage is well lit (see Figure 5.5a).

In the baseline situation, the agents in the crowd start applauding when the speech ends (see Figure 5.5b). After the applause, bright lights switch on that illuminate the room. The agents then walk out of the room (see Figure 5.5c).

In the emergency situation, the speech is interrupted by an explosion 18 seconds in. At the same time, the lights of the room and the scene switch off, then flicker on after a few seconds. At the same time the sound of a fire alarm starts sounding, and the sound of sirens and screams can be heard, coming from outside. There are three agents lying on the ground. The agents in the crowd stand in their spots, looking around with scared body language (see Figure 5.5d), then run out of the room (see Figure 5.5e). The evacuation of the room is fluid, with minimal clogging at the exits.

**Analysis** We want to study the effect of stress on behavior, which means that the emergency scenario in our application should be able to induce measurable stress on the user. We use physiological measurement devices to detect stress. Physiological measurements, especially heart rate and skin conductivity, have been proven to be correlated with stress levels [5].

For our application to be useful, additionally, the behavior of the user should be different in the emergency and baseline situations. We analyze spatial and gaze data from the participants in order to search for differences in their behavior.

# Chapter 6

# Results

In this chapter, we analyze the data we collected during our experiment and discuss our conclusions.

Our experiment was a pilot with only five participants. Consequently we perform only preliminary analysis of the data with exploratory aims. We have two main goals for our analysis.

In Section 6.1 we want to see if our application can be used to qualitatively reproduce the stress response of people within a crowd in an emergency situation. As explained in Section 2.3, stress is a substantial imbalance between demand (physical or psychological) and response capability, under conditions where failure to meet the demand has important consequences [40]. We use the known effects of stress on physiological processes to determine whether our application produces the intended stress response.

In Sections 6.2 and 6.3 we attempt to validate our hypothesis that we can see a difference in behavior between our baseline and emergency scenarios. If the experimental subject is immersed in the scene, we expect them to behave —qualitatively— like a person would behave in a real-world situation. Detecting a difference in behavior in our data would mean that the participant experienced our two scenarios differently. Moreover, it would mean that our setup can detect the changes in behavior and therefore allows us to study them.

In Section 6.2 we present and analyze spatial data related to the locomotion of the user. In Section 6.3 we present and analyze the user's gaze patterns.

We use MATLAB scripts to perform data analysis.

## 6.1 Physiological data

We want to be able to study the behavior of humans in crowds in situations of stress. In order to do this, our application should be able to cause a stress response in the experimental subjects.

We used physiological sensors to collect the users' galvanic skin response and heart rate during their participation in the experiment. In this section, we present and analyze the collected data.

In Section 6.1.1, we present our skin conductivity results. In Section 6.1.2, we present our heart rate results. In Section 6.1.3 we analyze our results and discuss our conclusions.

### 6.1.1 Galvanic Skin Response

The theory of GSR holds that the conductance of human skin increases with the activity of the sweat glands in it. Sweating increases in situations of fear and stress. Therefore, we can use measurements of the conductance of a subject's skin as a measure of their stress or psychological arousal [10].

In our experiment, we expect to see a rise in the GSR of the subject when the alarm sounds, and no such rise in the baseline trials. If this happens, it means that our emergency scenario can cause a physiological reaction in the experimental subjects.

As seen in Figure 6.1, the GSR value of all but one of the participants rose sharply in the emergency trial when the explosion occurred.

### 6.1.2 Heart rate

Heart rate increases with stress and fear [46]. We can then use a measure of heart rate to check whether our experimental subjects experience a stress response to the emergency situation we expose them to in our experiment.

As seen in Figure 6.2, we find that the heart rate of all the participants stays mostly stable during all the trials. Unlike what we expected, we don't find an increase of the heart rate in our emergency situation.

### 6.1.3 Discussion

In our GSR data, our results mostly coincide with what we expected. Four out of five participants experience a peak in GSR when the alarm sounds. The measurement additionally stays mostly constant for participants one and two in the baseline trials. However, for one of the participants, there isn't a peak in the emergency-scenario GSR, only a continuation of a constant rising trend

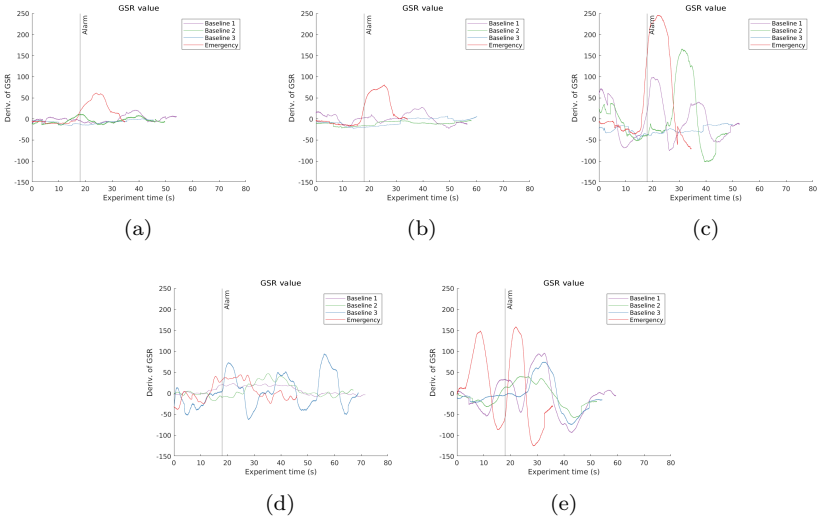



Figure 6.1: Derivatives of the galvanic skin response over time for the five participants

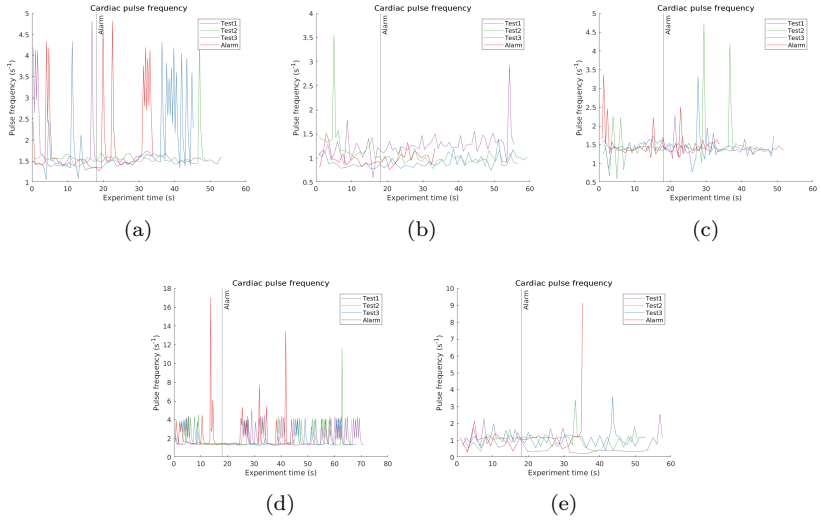



Figure 6.2: Pulse frequency over time for the five participants

that starts before the alarm. Participants 3 and 5 also have additional peaks in the baseline and emergency trials, which makes it harder to be sure that the alarm scenario peak is caused by the emergency.

Our pulse frequency results, on the other hand, don't show a rise in heart rate for any of the participants. This contrasts with the GSR results, which do seem to show a stress response. This may indicate that our application doesn't induce stress in the participants.

In the feedback given after their participation, some of the participants indicated that they hadn't felt scared because they had a good view of the exits, and the evacuation was fluid, without any clogging at the doors. Only participant 4 reported feeling scared by the emergency situation, which contrasts with their GSR measurements. Additionally, participant 4 reported developing motion sickness during immersion, which may alter their results, since motion sickness is correlated with lower feeling of presence.

The inconsistent results may be due to human variability in stress responses. They also may be a sign, especially given the heart rate results, that our application doesn't cause enough stress in the participant for the signal to be clear. It may also be the case that the participants only feel startled by the unexpected explosion, but don't feel the situation as stressful, maybe because they lack the feeling of presence, or because the emergency doesn't feel dangerous enough.

## 6.2 Spatial Data

If the participant in a VR experiment is immersed with a feeling of presence in the virtual scene, we expect them to behave —qualitatively— like a person would behave in a real-world situation. In the situation of the normal ending of a performance, we expect to see social behaviors like queueing and maintaning one's personal space, avoiding physical contact with others [48].

In an emergency situation, we expect to see those social behaviors preserved, since our scenario doesn't simulate the kind of emergency with a clear and immediate danger (like an out-of control fire inside the room) that could lead to panic-like behaviors [18].

In this section, do some exploratory analysis the locomotion patterns of the participants and the choices they make when exiting the room.

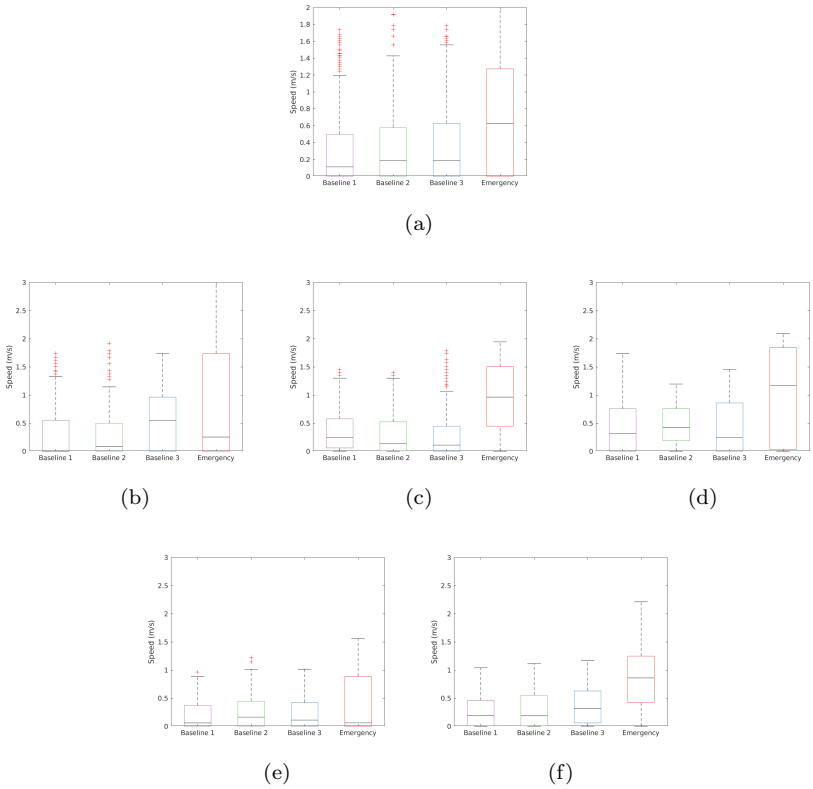
In Section 6.2.1, we analyze the speed at which the participants move when exiting the room. In Section 6.2.2 we look at when the participants start moving with relation to the timeline of events of the scenario. In Section 6.2.3 we look at the door the participants choose to exit the room through.

### 6.2.1 Navigation Speed

We analyze the speed at which the user moves when exiting the building. We expect to find that this speed increases in the emergency situation. We find

that the speed does seem higher, as seen in Figure 6.3. The average speeds for the four trials are $0.29m/s$, $0.32m/s$, $0.34m/s$, and $0.74m/s$, where the last one corresponds to the emergency trial. We consider the speeds starting when the crowd begins to move.

**Discussion** A higher navigation speed doesn't neessarily imply a higher arousal. We know that individuals in a crowd tend to follow the actions of the people around them [48]. Even if the subject doesn't experience the emergency situation as stressful and dangerous, they may move at an increased speed to match the people around them.




(a)

(b) (c) (d)

(e) (f)

Figure 6.3: (a) Speed aggregated over all the participants for all trials. (b-f) Speed per participant for all trials.

| | $\Delta t_1$ | $\Delta t_2$ | $\Delta t_3$ | $\Delta t_4$ |
|---|---|---|---|---|
| Participant 1 | 12.86 | 7.13 | 3.06 | 3.42 |
| Participant 2 | 1.33 | 1.77 | 3.23 | -9.01 |
| Participant 3 | 1.53 | 5.50 | 10.61 | 2.25 |
| Participant 4 | 7.83 | 3.00 | 1.98 | 8.58 |
| Participant 5 | 2.93 | 4.74 | 5.25 | -2.82 |
| Mean time | 5.30 | 4.43 | 4.82 | 0.48 |
| std of time | 4.45 | 1.88 | 3.08 | 5.97 |

Table 6.1: Table of the time between the first agent moving to leave the room and the participant starting to move

### 6.2.2 Time to Start Moving

We analyze the time that it takes for the subjects to start moving, in relation to the time the rest of the crowd starts moving.

In the baseline scenario, the crowd starts moving after they have finished applauding, at the same time that the lighting of the room changes from dark performance lighting to bright end-of-show lighting. This happens 33 seconds into the scenario. The end of the speech happens in second 25 of the scenario.

In the emergency scenario, when the explosion sounds the performance lights go out, and flicker back on after a few seconds. Meanwhile, an alarm starts sounding, and the sounds of sirens and screams start to sound outside the room. the crowd starts to run to evacuate after the lights come back on, on second 27. The explosion happens 18 seconds into the scenario.

As shown in Table 6.1, the experimental subjects generally started to move a few seconds after the crowd started to move. Nobody started to leave while the crowd was applauding.

In the emergency scenario, some of the participants started moving before the crowd started moving.

**Discussion** Participants moving before the crowd in the emergency scenario may indicate that they were anxious to evacuate the room. On the other hand, some participants reported being confused about whether the explosion and the sudden darkness were intended. This may have prompted them to test the application for a response.

With the two early movers removed, the average time to leave is similar to those in the other trials (4.75). This is expected, since previous studies have shown that, even if a fire alarm is sounding, people tend to wait and see how the people around them respond, rather than be the first to take action [52].

| | Baseline 1 | Baseline 2 | Baseline 3 | Emergency |
|---|---|---|---|---|
| Participant 1 | 1 | None | 1 | 3 |
| Participant 2 | 4 | 4 | 2 | 1 |
| Participant 3 | 1 | 4 | 4 | 1 |
| Participant 4 | 3 | 2 | 3 | 2 |
| Participant 5 | 4 | 4 | 4 | 3 |

Table 6.2: Door the participant used to exit the room

### 6.2.3 Exit Door Choice

As shown in Table 6.2, our subjects chose a door they hadn't previously used in all the emergency situations, and there's no obvious pattern to their door choices. Participant 1 exited through the wall in the Baseline 2 trial.

**Discussion** Previous studies have shown that people usually choose to exit a building through an exit that is familiar to them, such as the one they used to enter the building. If an emergency exit is usually barred from circulation, it's usually ignored in an emergency [49]. Since the subjects experience first the baseline non-emergency situation, and have time to familiarize themselves with the exits, we expect that they will tend to choose the same door in all the experiments, or at least choose a familiar exit in the emergency situation.

Our participants didn't conform to this pattern. This may be because all exits are close together in the room and clearly visible, making them interchangeable. The tendency to keep to familiar routes tends to increase in an emergency [52]. That the users chose a new exit in the emergency scenario may also be a sign that they weren't feeling stressed by the emergency.

## 6.3 Gaze

Using the FOVE headset allows us to track the gaze direction of the experimental subject. The ultimate goal of building our scenario, which falls beyond the scope of this project, is to use analysis of gaze data to study the behavior of individuals in dense crowds.

In this section, we analyze our preliminary data to see whether we can find differences in the gaze behavior between the baseline scenario and the emergency scenario.

In Section 6.3.1, we look at the difference between gaze fixations. In Section 6.3.2, we compare the direction of the subject's gaze to their trajectory and the

orientation of their body. In Section 6.3.3 we discuss our findings.

### 6.3.1 Angle between Fixations

We use MATLAB scripts to analyze the gaze data of the subjects and group the raw gaze data into a series of fixations on different points. The code for the computation of the fixations was developed by Florian Berton.

Following [38], we consider that a temporal series of gaze directions constitute a fixation when, for a temporal window of 80ms or more, all the gaze directions are within a range of 3.0 degrees from the initial direction.

We analyze the horizontal angles between each two successive fixations to measure how much the subject is looking around. We analyze the data after the speech has ended or the explosion has occurred, since that's when the scenarios start to differ.

As shown in Figure 6.4, we don't find a visible difference in the angles between fixations in the two scenarios.

### 6.3.2 Gaze vs Trajectory and Camera Rotation

In Section 6.3.1 we compare the directions of the subject's gaze among themselves. Here, we compare the direction of the gaze with both the trajectory of the navigation of the subject and the rotation of the camera, i.e. the orientation in space of the virtual body of the subject. Our aim here is also to detect a difference in gaze behavior between our two scenarios.

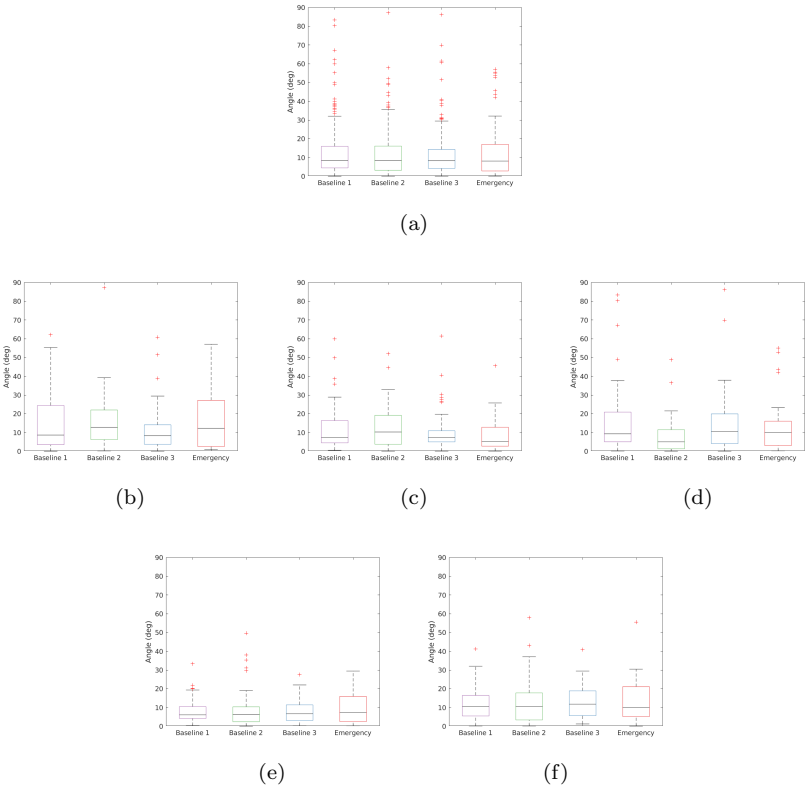
As shown in Figures 6.5 and 6.6, we don't find a visible difference in the angles between the gaze and the camera angle or the trajectory in the two scenarios.

### 6.3.3 Discussion

We didn't find a difference in the amount people looked around in our two scenarios by doing our analysis.

As far as we know, there's no scientific literature on the subject of gaze behavior in emergency situations. We therefore didn't know what we should expect for these results.

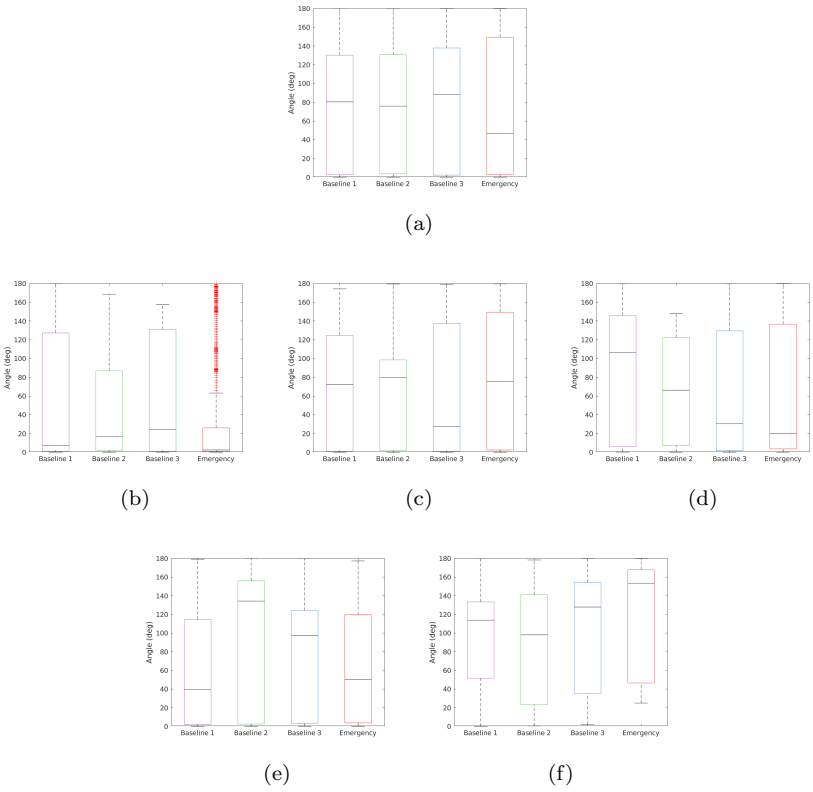
If there's a difference to be found, a reason that we may not have found it is our chosen method of navigation. Our users used a joystick for navigation. This is an unnatural way to represent locomotion, and may influence how much the person looks around, e.g. depending on the relative difficulty of turning around with a joystick vs. turning around while walking vs. looking around.




(a) (b) (c) (d) (e) (f)

Figure 6.4: (a) Angle between successive fixations aggregated over all the participants for all trials. (b-f) Angle changes per participant for all trials.

It's also possible that the difference in gaze behavior is not in how much the person looks around, but what the person looks at. We didn't collect data on gaze allocation, which requires non-trivial setup to do.

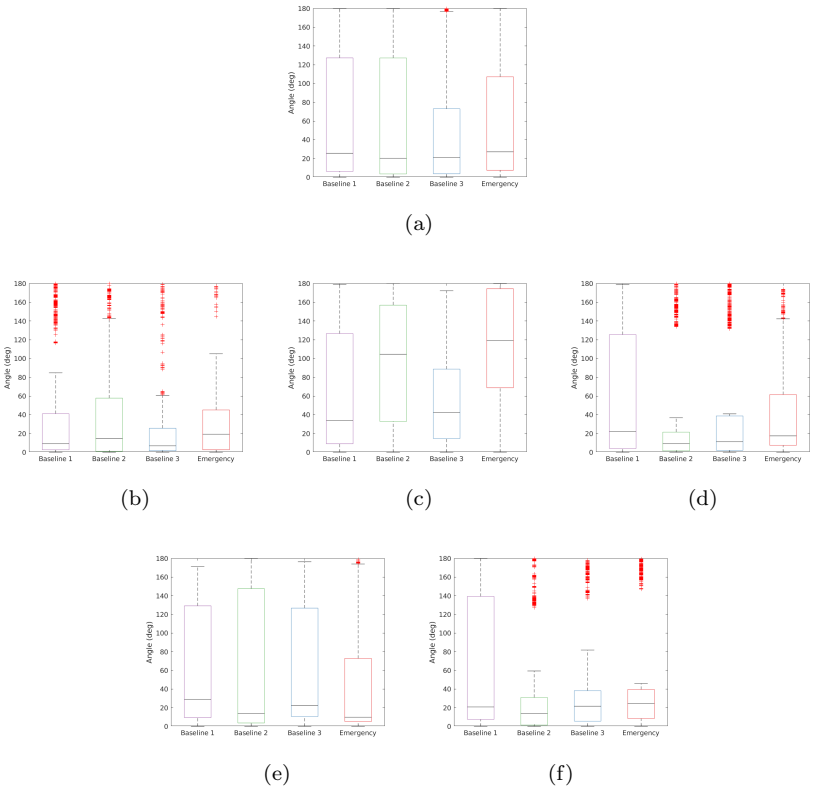



(a)

(b) (c) (d)

(e) (f)

Figure 6.5: (a) Difference between gaze and camera orientation aggregated over all the participants for all trials. (b-f) Angle difference per participant for all trials.




(a) (b) (c) (d) (e) (f)

Figure 6.6: (a) Difference between gaze and and trajectory orientation aggregated over all the participants for all trials. (b-f) Angle difference per participant for all trials.

# Chapter 7

# Conclusions and Future Work

## 7.1 Conclusions

The aim of this master's final project was to work on the ongoing research on human behavior in crowds using virtual reality and gaze analysis conducted by Julien Pettre et al. at INRIA. My contribution to that objective was to build and experimentally validate a virtual reality application that simulates an immersive scene where two versions of a scenario involving a dense crowd exiting a building can be simulated —one in which a performance happens and ends normally and one where the performance is interrupted by an emergency evacuation.

The application was built using the Unity3D game engine and an in-development platform for VR crowd simulation experiments.

The validation of the application was done with a pilot experiment with five participants. We analyzed the data from our experiment with two objectives: to investigate whether we can cause a stress reaction via immersion in our application, and to examine whether we can elicit and detect a difference in the behavior of the participants in our two scenarios.

We use two physiological markers of stress reaction: skin conductivity (sweat) and heart rate. We find that skin conductivity rises in the stress situation in all but one of our participants. By contrast, we don't find a heart rate increase in any of the participants. Some possible causes of our heart rate result could be the scenario not being stressful enough, the application not inducing enough feeling of presence.

We analyze the spatial data related to the movement of the participants. We find that the participants exited the room faster and started moving sooner in the emergency situation. Further analysis would be needed to determine the cause of this difference. We find that the choice of exit doors of the participants doesn't match previous research that shows that people favor familiar exits. A possible cause of this may be that the doors in our scene are too close together and can be seen from all points of the room, making them interchangeable.

Finally, we analyze the gaze behavior of the participants. We don't find a difference in the breadth of gaze directions in our two situations. Further analysis of this data could, for example, look at the gaze allocation patterns of the participants.

During the course of my stay at INRIA, aside from working on my project, I participated in the development of an open source platform for crowd simulation, ChAOS, written in C++. This required learning the mechanisms of collaboration in code development in a long-term project, which was a great opportunity that isn't frequent in the context of academic study.

## 7.2 Future Work

In the scope of this project, we built an application that can be used for the study of the behavior of individuals in dense crowds. We performed a pilot experiment, with only a few participants, that we used to validate the application.

Our findings from this pilot experiment should be used to improve our application. The immersiveness of the VR environment could be enhanced, e.g. by developing some mechanism of haptic feedback that enables the user to feel the collisions with the agents in the crowd, or by using a more immersive mechanism of locomotion. The floor plan of the room could be made more complex in order to make it harder for the participants to locate the exits. The signs of danger that the participants perceive in the emergency situation could be increased, e.g. by simulating some more immediate danger inside the room, or by creating some clogging in the exits.

More sophisticated data analysis and further experiments should be done to try to determine the cause of the changes in speed and time to move in our two scenarios. Our gaze data could be analyzed further to try to determine if there's a change in behavior that we can measure with our setup.

Designing and running experiments that can use the application as a tool in the study of human behavior in crowds is also future work.

# Bibliography


[1] RVO2 Library - Downloads.

[2] Unity - Manual: Global Illumination, 2017.

[3] Unity - Manual: Inner Workings of the Navigation System, 2017.

[4] Unity - Manual: Real-time lighting, 2017.

[5] Jennifer A. Healey and Rosalind W. Picard. Detecting Stress During Real-World Driving Tasks Using Physiological Sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, 2005.

[6] Junghyun Ahn, Stéphane Gobron, Quentin Silvestre, Horesh Ben Shitrit, Mirko Raca, Julien Pettré, Daniel Thalmann, Pascal Fua, and Ronan Boulic. Long term real trajectory reuse through region goal satisfaction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7060 LNCS, pages 412–423, 2011.

[7] Adrian F. Aveni. The Not-So-Lonely Crowd: Friendship Groups in Collective Behavior. *Sociometry*, 40(1):96, mar 1977.

[8] Tom Banton, Jeanine Stefanucci, Frank Durgin, Adam Fass, and Dennis Proffitt. The Perception of Walking Speed in a Virtual Environment. *Presence: Teleoperators and Virtual Environments*, 14(4):394–406, aug 2005.

[9] Benoit Bideau, Richard Kulpa, Nicolas Vignais, Sébastien Brault, and Franck Multon. Using Virtual Reality to Analyze Sports Performance. *IEEE Computer Graphics and Applications*, 2009.

[10] Wolfram. Boucsein. *Electrodermal activity*. Springer Science+Business Media, LLC, 2012.

[11] D. C. Brogan and N. L. Johnson. Realistic human walking paths. In *Proceedings - IEEE Workshop on Program Comprehension*, volume 2003-Janua, pages 94–101. IEEE Comput. Soc, 2003.

[12] Julien Bruneau. Simulation de la Panique. Technical report, SpirOps, Paris, 2011.

[13] Julien Bruneau. Studying and modeling complex interactions for crowd simulation. 2017.

[14] Julien Bruneau, Anne Hélène Olivier, and Julien Pettré. Going Through, Going Around: A Study on Individual Avoidance of Groups. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):520–528, apr 2015.

[15] Ujjal Chattaraj, Armin Seyfried, and Partha Chakroborty. Comparison of Pedestrian Fundamental Diagram Across Cultures. *Advances in Complex Systems*, 12(03):393–405, jun 2009.

[16] Alessandro Corbetta, Luca Bruno, Adrian Muntean, and Federico Toschi. High statistics measurements of pedestrian dynamics. In *Transportation Research Procedia*, volume 2, pages 96–104, 2014.

[17] Marco Costa. Interpersonal Distances in Group Walking. *Journal of Nonverbal Behavior*, 34(1):15–26, mar 2010.

[18] R. F. Fahy and G. Proulx. 'Panic' and human behaviour in fire. *Proceedings of the 4th International Symposium on Human Behaviour in Fire*, 2009.

[19] Brett R Fajen and William H Warren. Behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of experimental psychology. Human perception and performance*, 29(2):343–62, apr 2003.

[20] Philip W. Fink, Patrick S. Foo, and William H. Warren. Obstacle avoidance during walking in real and virtual environments. *ACM Transactions on Applied Perception*, 4(1):2–es, 2007.

[21] Paolo Fiorini and Zvi Shiller. Motion Planning in Dynamic Environments Using Velocity Obstacles. *The International Journal of Robotics Research*, 17(7):760–772, jul 1998.

[22] Susan Folkman. Stress: Appraisal and Coping. In *Encyclopedia of Behavioral Medicine*, pages 1913–1915. Springer New York, New York, NY, 2013.

[23] FOVE. FOVE — Eye Tracking Virtual Reality Headset, 2017.

[24] Maia Garau, Mel Slater, David-Paul Pertaub, and Sharif Razzaque. The Responses of People to Virtual Humans in an Immersive Virtual Environment. *Presence: Teleoperators and Virtual Environments*, 14(1):104–116, feb 2005.

[25] Martin Gérin-Lajoie, Carol L. Richards, Joyce Fung, and Bradford J. McFadyen. Characteristics of personal space during obstacle circumvention in physical and virtual environments. *Gait and Posture*, 27(2):239–247, 2008.

[26] E.A.D. Heide. Common Misconceptions about disasters: Panic, the "Disaster Syndrome," and Looting. In *The First 72 Hours: A Community Approach to Disaster Preparedness*, pages 340–380. 2004.

[27] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, may 1995.

[28] Claudia Hendrix and Woodrow Barfield. The sense of presence within auditory virtual environments. *Presence: Teleoperators and Virtual Environments*, 5(3):290–301, jan 1996.

[29] Bruno Herbelin. Virtual reality exposure therapy for social phobia. 3351:1–149, 2005.

[30] Hoffman HG, Chambers GT, Meyer W J 3rd, Arceneaux LL, Russell WJ, Seibel EJ, Richards TL, Sharar SR, Patterson DR, Hunter G Hoffman, Gloria T Chambers, Walter J 3rd Meyer, Lisa L Arceneaux, William J Russell, Eric J Seibel, Todd L Richards, Sam R Sharar, and David R Patterson. Virtual reality as an adjunctive non-pharmacologic analgesic for acute burn pain during medical procedures. *Annals of Behavioral Medicine*, 41(2):183–191, 2011.

[31] John H Hollman, Robert H Brey, Tami J Bang, and Kenton R Kaufman. Does walking in a virtual environment induce unstable gait? *Gait & Posture*, 26(2):289–294, 2007.

[32] JD Sime. The concept of panic. *Fires and human behavior*.

[33] Fiona (Psychologist) Jones, Jim. Bright, and Angela. Clow. *Stress : myth, theory and research*. Prentice Hall, 2001.

[34] Joshua Knapp and Jack Loomis. Visual Perception of Egocentric Distance in Real and Virtual Environments. In *Virtual and Adaptive Environments*, pages 21–46. 2003.

[35] Margrethe Kobes, Ira Helsloot, Bauke de Vries, and Jos G. Post. Building safety and human behaviour in fire: A literature review, 2010.

[36] Joung H Kwon, Chalmers Alan, Silvester Czanner, Gabriela Czanner, and John Powell. A Study of Visual Perception: Social Anxiety and Virtual Realism. In *Proceedings of the 25th Spring Conference on Computer Graphics*, pages 167–172, 2009.

[37] Xiao Dong Liu, Wei Guo Song, and Wei Lv. Empirical data for pedestrian counterflow through bottlenecks in the channel. In *Transportation Research Procedia*, volume 2, pages 34–42, 2014.

[38] SD Lynch, J Pettré, J Bruneau, R Kulpa ... IEEE Conference on ..., and Undefined 2018. Effect of virtual human gaze behaviour during an orthogonal collision avoidance walking task. In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 136–142, 2018.

[39] Hanspeter A. Mallot, Sabine Gillner, Hendrik A.H.C. Veen, and Heinrich H. Bülthoff. Behavioral experiments in spatial cognition using virtual reality. *Spatial Cognition*, pages 447—-467, 1998.

[40] JE McGrath. *Social and psychological factors in stress*. Oxford, 1970.

[41] Michael Meehan, Brent Insko, Mary Whitton, and Frederick P. Brooks. Physiological measures of presence in stressful virtual environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques - SIGGRAPH '02*, page 645, 2002.

[42] Merrian-Webster Inc. Stress — Definition of Stress by Merriam-Webster, 2018.

[43] Betty J. Mohler, Sarah H. Creem-Regehr, and William B. Thompson. The influence of feedback on egocentric distance judgments in real and virtual environments. In *Proceedings of the 3rd symposium on Applied perception in graphics and visualization - APGV '06*, page 9, 2006.

[44] Betty J Mohler, Sarah H Creem-Regehr, William B Thompson, and Heinrich H Bülthoff. The Effect of Viewing a Self-Avatar on Distance Judgments in an HMD-Based Virtual Environment. *Presence: Teleoperators and Virtual Environments*, 19(3):230–242, jun 2010.

[45] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5(4):e10047, apr 2010.

[46] V Mustonen and M Pantzar. Tracking social rhythms of the heart: from dataism to art. 2013.

[47] Anne Hélène Olivier, Antoine Marin, Armel Crétual, and Julien Pettré. Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. *Gait and Posture*, 36(3):399–404, 2012.

[48] Xiaoshan Pan, Charles S. Han, Ken Dauber, and Kincho H. Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & SOCIETY*, 22(2):113–132, oct 2007.

[49] Nuria Pelechano and Norman Badler. Modeling Crowd and Trained Leader Behavior during Building Evacuation. *IEEE Computer Graphics and Applications*, 26(6):80–86, nov 2006.

[50] Christopher Peters and Cathy Ennis. Modeling groups of plausible virtual pedestrians. *IEEE Computer Graphics and Applications*, 29(4):54–63, 2009.

[51] Julien Pettré, Jan Ondřej, Anne-Hélène Olivier, Armel Cretual, and Stéphane Donikian. Experiment-based modeling, simulation and validation of interactions between virtual walkers. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09*, page 189, 2009.

[52] G. Proulx. A Stress Modell for People Facing Fire. *Journal of Environmental Psychology*, 13:137–147, 1993.

[53] Rebekka S. Renner, Boris M. Velichkovsky, and Jens R. Helmert. The perception of egocentric distances in virtual environments - A review. *ACM Computing Surveys*, 46(2):1–40, 2013.

[54] Craig W Reynolds. Steering Behaviors For Autonomous Characters. In *Proceedings of Game Developers Conference*, 1999.

[55] Roger L. Huges. The Flow of Human Crowds. *Annual Review of Fluid Mechanics*, 35:169–82, 2003.

[56] Martijn J. Schuemie, Peter van der Straaten, Merel Krijn, and Charles A.P.G. van der Mast. Research on Presence in Virtual Reality: A Survey. *CyberPsychology & Behavior*, 4(2):183–201, apr 2001.

[57] Mel Slater, Amela Sadagic, Martin Usoh, and Ralph Schroeder. Small-Group Behavior in a Virtual and Real Environment: A Comparative Study. *Presence: Teleoperators and Virtual Environments*, 9(1):37–51, feb 2000.

[58] Mel Slater, Martin Usoh, and Anthony Steed. Depth of Presence in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 3(2):130–144, jan 1994.

[59] Jonathan Steuer. Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, 42(4):73–93, dec 1992.

[60] Adrien Treuille, Seth Cooper, Zoran Popović, Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. In *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, volume 25, page 1160, New York, New York, USA, 2006. ACM Press.

[61] Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE, may 2008.

[62] Wouter van Toll, Roy Triesscheijn, Marcelo Kallmann, Ramon Oliva, Nuria Pelechano, Julien Pettré, and Roland Geraerts. A comparative study of navigation meshes. In *Proceedings of the 9th International Conference on Motion in Games - MIG '16*, pages 91–100, 2016.

[63] Peter Willemsen, Mark B. Colton, Sarah H. Creem-Regehr, and William B. Thompson. The effects of head-mounted display mechanics on distance judgments in virtual environments. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization - APGV '04*, 2004.

[64] Bob G. Witmer and Michael J. Singer. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3):225–240, jun 1998.

[65] David Wolinski. *Microscopic crowd simulation : evaluation and development of algorithms*. PhD thesis, jan 2016.

[66] Beibei Zhan, Dorothy N. Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li Qun Xu. Crowd analysis: A survey. *Machine Vision and Applications*, 19(5-6):345–357, oct 2008.

[67] B Zhou, X Tang, and X Wang. Measuring Crowd Collectiveness. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3049–3056, 2013.

[68] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents.

In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2871–2878, 2012.